Sinkhole Attack Identification using Hierarchial Intrusion Detection System

Dissertation submitted to
Visvesvaraya National Institute of Technology, Nagpur
in partial fulfillment of the requirements for the award of
the degree

Master of Technology
in
Computer Science & Engineering

by

BHAVESH DNYANESHWAR KHANDAR (MT23MCS008)

under the guidance of

Dr. D.D.Shrimankar

Department of Computer Science & Engineering

Visvesvaraya National Institute of Technology

Nagpur 440 010 (India)

DECEMBER 2025

Sinkhole Attack Identification using Hierarchial Intrusion Detection System

Dissertation submitted to
Visvesvaraya National Institute of Technology, Nagpur
in partial fulfillment of the requirements for the award of
the degree

Master of Technology
in
Computer Science & Engineering

by

BHAVESH DNYANESHWAR KHANDAR (MT23MCS008)

under the guidance of

Dr. D.D.Shrimankar

Department of Computer Science & Engineering

Visvesvaraya National Institute of Technology

Nagpur 440 010 (India)

DECEMBER 2025

Declaration

I, **Bhavesh Dnyaneshwar Khandar**, hereby declare that this dissertation titled **" Sinkhole Attack Identification using Hierarchial Intrusion Detection System"** is carried out by me in the Department of Computer Science & Engineering of Visvesvaraya National Institute of Technology, Nagpur. The work is original and has not been submitted earlier whole or in part for the award of any degree at this or any other Institution.

BHAVESH DNYANESHWAR KHANDAR

Name and signature Khandar

Date: 09/12/25

Certificate

This to certify that the dissertation titled **"Sinkhole Attack Identification using Hierarchial Intrusion Detection System"** is submitted by **Bhavesh Dnyaneshwar Khandar** in partial fulfillment of the requirements for the award of the degree of **Master of Technology in Computer Science & Engineering**, VNIT Nagpur. The work is comprehensive, complete, and fit for final evaluation.

Deepti 9/12/25

Dr. D.D. Shrimankar
Department of Computer Science & Engineering, VNIT Nagpur

Dr. Umesh A. Deshpande
Head, Department of Computer Science & Engineering, VNIT Nagpur
Date:

External Examiner

Date:

Department of Computer Science & Engineering
Visvesvaraya National Institute of Technology, Nagpur

Non-Plagiarism Certificate

Certified that M. Tech Dissertation titled **"Sink- hole Attack Identification using Hierarchial Intrusion Detection System"** submitted by **Bhavesh Dnyaneshwar Khandar**, Enrollment No. **MT23MCS008**, ID No. **30133** has been checked for plagiarism using TURNITIN software and the overlap is found within prescribed limits. A summary of the report is given below.

ORIGINALITY REPORT

21%	13%	18%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

Khandar 09/12/25

Bhavesh Dnyaneshwar Khandar

Enrollment No: MT23MCS008

9/12/25

Dr. D.D.Shrimankar

Department of Computer Science & Engineering

VNIT, Nagpur

ACKNOWLEDGEMENT

I would like to express my deepest gratitude and sincere appreciation to my supervisor, **Dr. D.D.Shrimankar**, for providing me with the opportunity to work on this thesis, Sink-hole Attack Identification using Hier- archial Intrusion Detection System. Their unwavering support, insightful guidance, technical expertise, and constant encouragement have been instrumental in shaping this research. I feel truly privileged to have had the opportunity to work under their mentorship, which has not only enriched my understanding of network security but also inspired me to push the boundaries of innovation.

I extend my heartfelt thanks to **Dr. Umesh A. Deshpande**, Head of the Department of Computer Science and Engineering, for their invaluable support and guidance throughout this academic journey. Their encouragement and belief in my potential have been pivotal in the successful completion of this project, and I am deeply grateful for their contributions.

I would also like to express my gratitude to all the teaching and non-teaching faculty members of the Department of Computer Science and Engineering at Visvesvarya National Institute of Technology for their knowledge sharing, constructive feedback, and provision of essential resources whenever needed. Their assistance has been vital in ensuring the smooth progress of this research.

Finally, I owe a special thanks to my family and friends for their unwavering support, love, and encouragement throughout this journey. Their belief in my abilities, patience during my long hours of work, and motivation during challenging times have been the cornerstone of my success. This thesis is as much a reflection of their support as it is of my efforts.

Bhavesh Dnyaneshwar Khandar

ABSTRACT

Sinkhole attacks present a threat, to the security of 6LoWPAN networks. These attacks occur when a malicious node is set up by an attacker pretending to have a hop count to the sink node. This false information leads network nodes to route their traffic through the node allowing the attacker to manipulate the data. The proposed approach is validated on the public benchmark datasets, CIC-IDS-2017 and CSE-CIC-IDS-2018. Results demonstrate that our proposed approach besides effective and robust zero-day detection, outperforms both the baseline and existing approaches, achieving high classification performance, up to 96 percent balanced accuracy. Additionally, the proposed approach is easily adaptable without any retraining and takes advantage of n-tier deployments to reduce bandwidth and computational requirements while preserving privacy constraints. The suggested method combines anomaly detection and multi class classification using a model to identify sinkhole attacks. Initially anomaly detection analyzes network traffic patterns. Routing metrics to detect deviations that could signal sinkhole attacks. Subsequently the multi class classification stage categorizes identified anomalies, such as sinkhole attacks with learning techniques trained on labeled data. If this proposed strategy proves effective it could become a tool in safeguarding 6LoWPAN networks, against sinkhole attacks.

Abstract

Sinkhole attacks present a threat, to the security of 6LoWPAN networks. These attacks occur when a malicious node is set up by an attacker pretending to have a hop count to the sink node. This false information leads network nodes to route their traffic through the node allowing the attacker to manipulate the data. The proposed approach is validated on the public benchmark datasets, CIC-IDS-2017 and CSE-CIC-IDS-2018. Results demonstrate that our proposed approach besides effective and robust zero-day detection, outperforms both the baseline and existing approaches, achieving high classification performance, up to 96 percent balanced accuracy. Additionally, the proposed approach is easily adaptable without any retraining and takes advantage of n-tier deployments to reduce bandwidth and computational requirements while preserving privacy constraints. The suggested method combines anomaly detection and multi class classification using a model to identify sinkhole attacks. Initially anomaly detection analyzes network traffic patterns. Routing metrics to detect deviations that could signal sinkhole attacks. Subsequently the multi class classification stage categorizes identified anomalies, such as sinkhole attacks with learning techniques trained on labeled data. If this proposed strategy proves effective it could become a tool in safeguarding 6LoWPAN networks, against sinkhole attacks.

Contents

List of Figures **5**

1 Introduction **1**
1.1 Background 1
1.2 Existing sinkhole attack detection techniques and their limitations 1
1.3 The Multistage Hierarchical Intrusion detection system learning approach for catching of sinkhole attack 2
1.4 Proposed approach Overview 3
1.5 Contributions of the proposed work 3

1

2 Litrature Survey **5**

3 Proposed Approach **17**
3.1 System Architecture 17
3.2 Feature Extraction 18
3.3 Anomaly Detection Stage 18
3.4 Attack Classification Stage 18
3.5 Zero-Day Attack Detection 19
3.6 Extension Stage 19
3.7 Threshold Optimization 19
3.8 Implementation Details 20
3.9 Evaluation Metrics 20

4 Results and Discussion **21**

5 Conclusion and Future work **23**

References **25**

List of Figures

1.1 IoT devices Growth Chart rate with year 1
1.2 Sink hole attack senario 2

2.1 RPL Protocol : A review 6
2.2 Confusion martics Lables 8
2.3 Architecture for Multistage IDS 13
2.4 Visualization of all the stages and final multi stage approach of used train, validation and test strategy. 15

4.1 Multi class Classification 22

Chapter 1

Introduction

1.1 Background

A sinkhole attack is a type of cyber attack where the attacker creates a malicious node that advertises itself as having a short hop count to the sink node or some other critical node in the network.

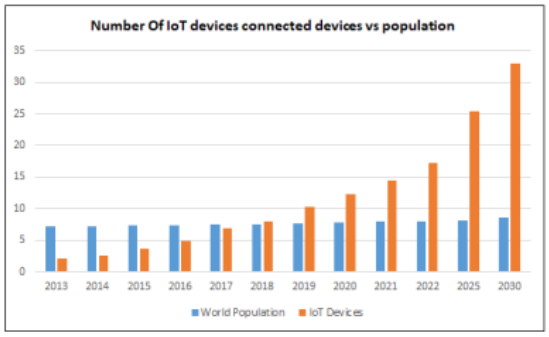

Fig. 1.1 IoT devices Growth Chart rate with year

Other nodes in the network are then fooled into routing their traffic through the malicious node, which can then be intercepted or modified by the attacker.

1.2 Existing sinkhole attack detection techniques and their limitations

Existing sinkhole attack detection techniques often rely on traditional machine learning algorithms. However, these algorithms are not well-suited for detecting sinkhole attacks in IoT networks due to the unique constraints of these networks, such as limited resources and dynamic topologies.

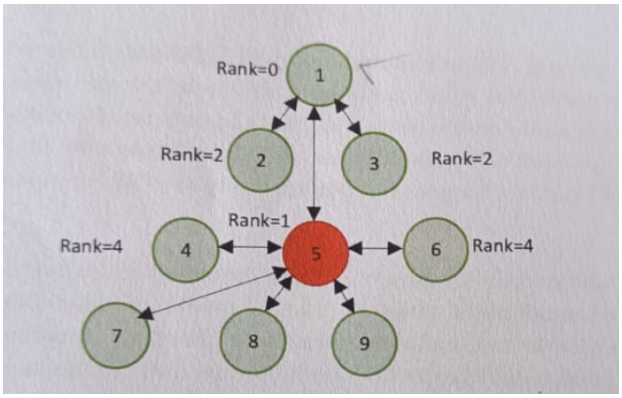

Fig. 1.2 Sink hole attack senario

Some of the limitations of traditional machine learning algorithms for sinkhole attack detection in IoT networks include High overhead: Traditional machine learning algorithms can be computationally expensive to train and deploy, which can be a problem for IoT devices with limited resources. Lack of adaptability: Traditional machine learning algorithms are trained on a specific dataset of data. Algorithm will not able to detect sinkhole attack effectively if there is any modification or change in the data.

1.3 The Multistage Hierarchical Intrusion detection system learning approach for catching of sinkhole attack

Sinkhole attacks have a threat to the security of today's networks, technologies like 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks). In a sinkhole attack, malicious nodes attract legitimate network traffic towards themselves by advertising false shortest path to a central sink node. Once the traffic is redirected, the attacker can , manipulate the data, leading to potential harm of confidentiality and integrity. Now, 6LoWPAN networks are definitely making IPv6 communication possible over low-power, low-bandwidth wireless networks, which is suitable for Internet of Things (IoT) and sensor applications. These networks come with limited processing power, memory, and energy, and they're built to work in difficult environments. It is outmost important to secure 6LoWPAN networks against different kinds of attacks, including sinkhole attacks, to keep IoT setups running securely.

But here is the tough decision part, the normal ways of detecting sinkholes have a hard time when it comes to 6LoWPAN networks. The unique features of 6LoWPAN setups, like their ever-changing network layouts, devices with limited resources, and the lack of a central monitoring and management system, make it hard for traditional detection methods to tell the good from the bad. So, we have this Hierarchical Intrusion Detection Systems (HIDS). It offers a reliable way to tackle the intricacies of identifying sinkhole attacks in 6LoWPAN networks. HIDS uses various numbers of detection stages and different detection tricks and sensor data to catch any abnormal packets and possible security risks. Additionally, the way HIDS is set up means it can adapt to the unique needs of 6LoWPAN networks and keep them safe from the ever-changing cyber threats issues.

1.4 Proposed approach Overview

The approach for sinkhole attack detection in 6LoWPAN using deep learning uses a convolutional neural network (CNN) to analyze network traffic patterns and find anomalous patterns that are susceptible of sinkhole attacks. The CNN is trained on a dataset of real-world 6LoWPAN network traffic traces, which makes it robust to variations in network traffic patterns and RPL routing metrics. The proposed approach is also lightweight and energy-efficient, which makes it convenient for using on IoT devices with constrained resources.

1.5 Contributions of the proposed work

The proposed work makes the following contributions It is the first to use deep learning for sinkhole attack detection in 6LoWPAN networks. It proposes a lightweight and energy-efficient deep learning model for sinkhole attack detection that is good for deployment on IoT devices with constrained resources. It demonstrates that the proposed approach can have high accuracy in knowing sinkhole attacks, even in the presence of noise and uncertainty in the data. Benefits of the proposed approach for saving 6LoWPAN networks from sinkhole attacks The proposed approach can help to saving 6LoWPAN networks from sinkhole attacks in the following ways: It can help to detect sinkhole[ASRNF13] attacks initially, before they can cause havey damage. It can help to reduce the overhead cased which are present in existing techniques. It can help to improve the security of 6LoWPAN networks, making them more power full to sinkhole attacks.

Chapter 2

Litrature Survey

The Internet of Things (IoT) is a changing technology that integrates physical objects into the internet, empowering them to collect and exchange data. IoT applications leads to various domains, developing smart and trusty healthcare, smart vehicles, smart cities, smart grids, and smart homes. Each is equipped with intelligent devices that communicate with the majority of devices via a gateway node over the internet.

IoT-based [MBC20] applications play an important role in modern societies, facilitating the monitoring and control of remote IoT devices. Resource-constrained devices, such as sensor nodes, can connect to IPv6 using Low-Power Wireless Personal Area Networks (6LoWPAN), where each node has its own IPv6 address. 6LoWPAN is considered a standard protocol for realizing IPv6 communication on wireless networks and is preferred for IoT implementation.

IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) is a standardized routing protocol for IoT devices. In this protocol, a sink node acts as a root, connecting to the internet via a gateway. To establish routes, RPL constructs a Destination-Oriented Directed Acyclic Graph (DODAG) tree, with one root and multiple nodes joining the network as its children. Four ICMPV6 control messages are employed for building routing topology and updating routing information: DODAG Information Object (DIO), Destination Advertisement Object (DAO), Destination Advertisement Object Acknowledgment (DAO-ACK), and DODAG Information Solicitation (DIS).

The emergence of new technologies often brings along security concerns and vulnerabilities. IoT [TLY+23] is no exception, and its widespread adoption has introduced numerous challenges in terms of attacks and defense strategies. This technology faces unique threats not encountered in traditional networks. Several security challenges in IoT include:

IoT networks are vulnerable to various attacks that can disrupt communication and compromise data security. Various types of attacks pose threats to networks, such as blackhole attacks, wherein malicious nodes intentionally discard all incoming packets, thereby causing a disruption by [SSS18]silencing other nodes; selective forwarding attacks, where affected nodes selectively drop packets disrupting communication; sinkhole attacks, where malicious nodes advertise incorrect routing information attracting traffic to themselves allowing them to capture, modify, or drop packets at will; wormhole attacks, where malicious nodes create a tunnel between two points in the network enabling them to intercept and tamper with data flowing through the tunnel; and Sybil attacks, where attackers forge iden-

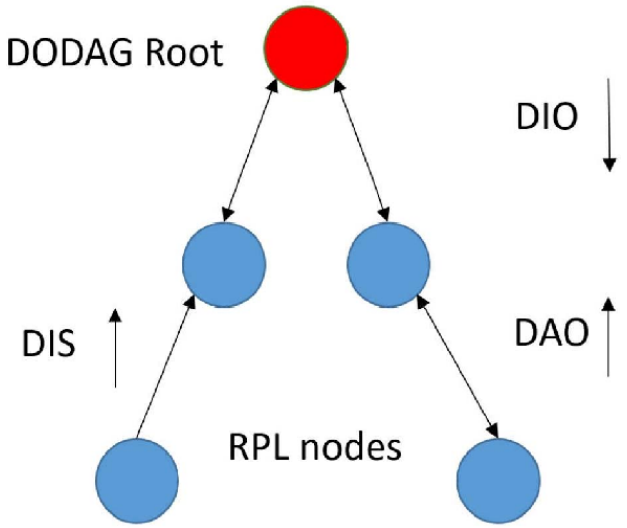

Fig. 2.1 RPL Protocol : A review

tities impersonating legitimate nodes to gain unauthorized access to resources or disrupt network operations.

Among these attacks, sinkhole attacks pose a particularly significant threat in wireless networks. Sinkhole nodes falsely advertise a low hop count to the sink node, tricking other nodes into routing their traffic through them. This enables the attacker to intercept, modify, or drop packets. Figure 3 illustrates the sinkhole attack scenario.

If a sinkhole node successfully infiltrates the network, it can adversely affect the network in three ways:

1. Packet deletion: The sinkhole node may simply discard received packets, causing data loss.

2. Packet modification: The sinkhole node may alter the contents of packets, introducing errors or malicious code.

3. Packet delay: The sinkhole node may intentionally delay packets, disrupting communication and degrading network performance.

Researchers have developed various techniques to detect sinkhole attacks, which are malicious nodes that disrupt communication in IoT networks. These techniques can be divided into three main categories:

- Reputation-based methods: These methods use the reputation of nodes to identify malicious ones. For instance, the Secure-RPL protocol tracks the rank values of nodes and employs a threshold function and hash chain authentication technique to detect sinkhole attacks.

- Trust-based methods: These methods determine the trustworthiness of nodes using trust values. For example, the Trust-RPL protocol calculates the trustworthiness

of nodes based on their recommended and direct trust values. Nodes possessing elevated trust values are selected for secure routing, whereas nodes with diminished trust values are classified as either exhibiting selfish behavior or engaging in malicious activities.

- Anomaly-based methods: These methods detect sinkhole attacks by identifying unusual behavior in the network. For instance, the SoS-RPL model uses rate and rank information to detect sinkhole attacks, while the INDRES method employs watchdogs to monitor nodes for suspicious activity.

These techniques help protect IoT networks from sinkhole attacks and maintain secure communication.

Some papers also proposed some good methods to do so and the method is explained below:

NADSA is a method for detecting sinkhole attacks. It consists of three phases:

- Phase 1: This phase calculates the minimum possible hop count between the root node and the receiver. If the sender's hop count is less than the minimum possible hop count, then it is a malicious node.

- Phase 2: This phase keeps track of the number of DIO messages sent by each node. If a node's counter list of neighbors shows that a node has sent more DIO messages than it has received, then it is a malicious node.

- Phase 3: This phase uses a fuzzy system to identify malicious nodes. The fuzzy system takes into account three parameters: Received Signal Strength Indicator (RSSI), Expected Transmission Count (ETX), and D. RSSI is a measure of the strength of a radio signal, ETX is the expected transmission count for each packet, and D is the difference between the sender's counter and the receiver's counter list of neighbors. The fuzzy system outputs a value that indicates whether or not the sender is a malicious node.

NADSA is an effective method for detecting sinkhole attacks in IoT networks. It is able to detect malicious nodes with high accuracy and low overhead.

The authors developed NADSA within the Contiki-NG operating system, specifically designed for the Internet of Things (IoT). Contiki OS facilitates the implementation and assessment of low-power, lossy networks, utilizing the RPL routing protocol for resource-constrained IoT devices. NADSA's primary focus is on detecting sinkhole attacks, yet its implementation involves the use of RPL in Contiki-NG. The Cooja simulator, compatible with Contiki, was employed to provide realistic results, particularly excelling in simulations involving 200 resource-constrained devices[HNLL06].

To assess NADSA, the authors conducted experiments, simulating a sinkhole attack against RPL. Various scenarios were tested, including topologies with 10, 30, and 60 normal nodes, each containing 10% malicious nodes. The simulation parameters are detailed in Table 3, and Figure 9 illustrates a 10-node topology. NADSA's performance was evaluated using specific parameters and compared against existing models like INTI, SoS-RPL, and RFTRUST.

In summary, the authors implemented NADSA in Contiki-NG, focusing on sinkhole attack detection using RPL. Cooja, a compatible simulator, was employed to achieve realistic results in simulations. The evaluation involved scenarios with normal and malicious nodes, with performance compared to existing models INTI, SoS-RPL, and RFTRUST, based on specific parameters.

True Positive Rate (TPR): This metric indicates the proportion of malicious nodes that have been successfully detected. It is calculated as the ratio of true positives (TP) to the total number of malicious nodes (TP + FN), expressed as a percentage.

True Positive (TP) represents the number of malicious nodes accurately identified by the detection system, while False Negative (FN) refers to the number of malicious nodes that have gone undetected.

False Positive Rate (FPR): This metric measures the frequency at which normal nodes are mistakenly classified as malicious nodes. It is calculated as the ratio of false positives (FP) to the total number of normal nodes (TN + FP), expressed as a percentage.

False Positive (FP) represents the number of normal nodes incorrectly identified as malicious nodes, while True Negative (TN) represents the number of normal nodes correctly classified as normal nodes.

	Predicted True	Predicted False
Actual True	True Positive(TP)	False Negative(FN)
Actual False	False Positive(FP)	True Negative(TN)

Fig. 2.2 Confusion martics Lables

Packet Delivery Ratio (PDR):

This metric quantifies the proportion of data packets successfully received at the receiver node out of the total number of data packets transmitted by the source node during the simulation. It is calculated using Equation 8:

```
PDR = (Received Packets / Sent Packets) * 100
```

where:

Received Packets: The total number of data packets successfully received at the receiver node. Sent Packets: The total number of data packets transmitted by the source node.

End-to-End Delay (E2ED):

This metric represents the time required for a packet to traverse the network from the sender node to the receiver node. It is measured in milliseconds (ms) and represents the overall latency of the network.

To evaluate the performance of NADSA in terms of packet delivery ratio (PDR) and end-to-end delay (E2ED), the proposed method was compared with the following scenarios:

1. Normal RPL scenario: The network operates under the standard RPL routing protocol.

2. Sinkhole attack scenario: The network is subjected to a sinkhole attack, where malicious nodes disrupt communication by advertising incorrect routing information.

3. NADSA with sinkhole attack: The network employs the proposed NADSA method to detect and mitigate sinkhole attacks.

In terms of true positive rate (TPR) and false positive rate (FPR), the proposed method was compared with three existing methods: INTI, SOS-RPL, and RFTRUST. TPR measures the proportion of malicious nodes successfully detected, while FPR indicates the frequency at which normal nodes are misidentified as malicious.

The results of the performance evaluation demonstrate that NADSA outperforms the other methods in terms of both PDR and E2ED. Additionally, NADSA exhibits superior TPR and lower FPR compared to the other methods, indicating its effectiveness in accurately identifying malicious nodes while minimizing false alarms.

Packet Delivery Rate Analysis:

The packet delivery rate (PDR) represents the proportion of packets successfully delivered to their intended destinations out of the total number of transmitted packets. The performance of various scenarios and methods was evaluated in terms of PDR under conditions of extreme network lossiness and sparsity. For packet delivery analysis, normal RPL exhibited a higher delivery rate compared to other methods. Figure 10 illustrates the comparison of normal RPL, sinkhole attack scenarios, NADSA, RFTRUST, SOS-RPL, and INTI. Normal RPL achieved a delivery rate of 81% across different numbers of nodes. The proposed method yielded delivery rates of 68%, 70%, and 73%. NADSA's higher delivery rate is attributed to its incorporation of additional parameters such as RSSI and ETX in the fuzzy logic system. These parameters enable the method to distinguish between network attack situations and lossy or sparse conditions. INTI, SoS-RPL, and RFTRUST rely on a large number of nodes to establish robust and stable communication between nodes; consequently, their performance deteriorates in an extremely lossy and sparse network.

End-to-End Delay Analysis:

End-to-end delay analysis assesses the average duration necessary for a packet to travel from its point of origin to its designated destination. The value of this parameter increases in the presence of network misbehavior. Packet retransmission is a key factor contributing to increased end-to-end delay during sinkhole attacks and in lossy or sparse networks. Figure 11 depicts the end-to-end delay for NADSA, normal RPL, sinkhole attacks, INTI, SOS-RPL, and RFTRUST models. In normal RPL, the RPL routing protocol operates without any intrusions, achieving delays of 82msec, 72msec, and 60msec. In the sinkhole attack scenario, where a malicious node is present without any prevention mechanism, the worst end-to-end delay outcomes are observed. NADSA achieved delays of 91msec, 81msec, and 71msec. The proposed method effectively detects malicious nodes and prevents their selection as parent nodes, thereby reducing packet retransmission and end-to-end delay.

True Positive Rate Analysis:

True positive rate (TPR) indicates the proportion of correct detections of malicious nodes during testing. Figure 12 demonstrates that the proposed method achieves true

positive rates of 89%, 83%, and 80%. Other methods fail to consider extremely lossy and sparse networks and, as a result, cannot achieve TPRs comparable to NADSA. The proposed method utilizes ETX 254 and RSSI to identify malicious nodes and differentiate between attack situations and extra lossy or sparse conditions, leading to its superior TPR.

False Positive Rate Analysis:

False positive rate (FPR) represents the proportion of incorrect detections of malicious nodes among all decisions made during testing of legitimate nodes. Figure 13 indicates that NADSA achieves FPRs of 24%, 28%, and 33%. These findings suggest that the proposed method exhibits a lower FPR compared to INTI, SOS-RPL, and RFTRUST. The proposed method incorporates various parameters to detect sinkhole attacks even under conditions of extreme network lossiness and sparsity. However, other methods do not focus on the probability of extra lossy and sparse networks, leading to mistakes in sinkhole attack detection. RFTRUST has a mechanism that signals the presence of sinkhole attacks even in extremely lossy and sparse networks, but its overall FPR remains higher than that of NADSA.

The evaluation of packet delivery rate (PDR), representing the successful delivery of packets to their intended destinations, was conducted under conditions of high network lossiness and sparsity. Normal RPL exhibited a superior delivery rate compared to other methods, achieving 81% across various node configurations. In comparison, the proposed NADSA method demonstrated delivery rates of 68%, 70%, and 73%, outperforming other scenarios. NADSA's enhanced delivery rate is attributed to its incorporation of additional parameters, such as RSSI and ETX, in the fuzzy logic system. These parameters enable the method to distinguish between network attack situations and lossy or sparse conditions. Conversely, INTI, SoS-RPL, and RFTRUST, relying on a large number of nodes for communication stability, exhibited deteriorating performance in extremely lossy and sparse networks.

End-to-End Delay Analysis:

The analysis of end-to-end delay, measuring the average time for packet transmission, revealed that in normal RPL, without intrusions, delays were 82msec, 72msec, and 60msec. In sinkhole attack scenarios, where malicious nodes were present without prevention mechanisms, NADSA effectively reduced end-to-end delays to 91msec, 81msec, and 71msec. By detecting malicious nodes and preventing their selection as parent nodes, NADSA mitigated packet retransmission and consequent delays. The figure comparing NADSA, normal RPL, sinkhole attacks, INTI, SOS-RPL, and RFTRUST illustrates these outcomes.

True Positive Rate Analysis:

The proposed method, NADSA, demonstrated superior true positive rates (TPR) of 89%, 83%, and 80%, indicating accurate detection of malicious nodes in testing. Other methods fell short in addressing extremely lossy and sparse networks, resulting in lower TPRs. NADSA's utilization of ETX 254 and RSSI facilitated the identification of malicious nodes and the differentiation between attack situations and extra lossy or sparse conditions, contributing to its higher TPR.

False Positive Rate Analysis:

NADSA exhibited lower false positive rates (FPR) of 24%, 28%, and 33% compared to INTI, SOS-RPL, and RFTRUST. The incorporation of various parameters in NADSA

allowed for the detection of sinkhole attacks even in extremely lossy and sparse networks. In contrast, other methods overlooked the probability of extra lossy and sparse networks, leading to errors in sinkhole attack detection. While RFTRUST signaled the presence of sinkhole attacks in such conditions, its overall FPR remained higher than that of NADSA.

The realm of the Internet of Things (IoT) comprises web-enabled smart devices fostering communication between physical entities, introducing a transformative trend. However, this innovation poses considerable security challenges, with sinkhole attacks emerging as a disruptive force within IoT networks. Despite the existence of intrusion detection methods targeted at sinkhole attacks, their efficacy diminishes in networks characterized by extra lossiness and sparsity. This research introduces a novel approach called NADSA, featuring two sections dedicated to identifying and isolating malicious nodes. Simulation results demonstrate that NADSA outperforms existing methods, including INTI, SOS-RPL, and RFTRUST, across key performance metrics: Packet Delivery Rate (PDR) (68%, 70%, and 73%), End-to-End Delay (E2ED) (81msec, 72msec, and 60msec), True Positive Rate (TPR) (89%, 83%, and 80%), and False Positive Rate (FPR) (24%, 28%, and 33%). While INTI, SOS-RPL, and RFTRUST exhibit high performance in dense node configurations with minimal packet retransmission, their effectiveness declines in extra lossy and sparse networks. In contrast, NADSA leverages distance measurement, DIO count, ETX, and RSSI, yielding commendable results in detecting sinkhole attacks within challenging network conditions.

From the above study I analysed the problem and the solution so by using this I will able to make machine learning modles.

A number of different approaches have been proposed for detecting sinkhole attacks in 6LoWPAN networks. These approaches can be broadly categorized into the following groups:

- Threshold-based approaches: These approaches use thresholds on network traffic or RPL routing metrics to identify anomalous behavior that is indicative of a sinkhole attack.
- Statistical approaches: These approaches use statistical techniques to model normal network traffic and identify deviations from the norm that are indicative of a sinkhole attack.
- Machine learning approaches: These approaches use machine learning algorithms to learn from labeled data and identify patterns in network traffic or RPL routing metrics that are indicative of a sinkhole attack.

Deep Learning Approaches

Deep learning is a type of machine learning that has recently emerged as a promising approach for detecting sinkhole attacks in 6LoWPAN networks. Deep learning algorithms are able to learn complex patterns in data, which makes them well-suited for identifying anomalous patterns that are indicative of sinkhole attacks.

A number of different deep learning architectures have been proposed for sinkhole attack detection in 6LoWPAN networks. These architectures include convolutional neural networks (CNNs), recurrent neural networks (RNNs), and autoencoders.

Evaluation Metrics

The performance of sinkhole attack detection algorithms is typically evaluated using the following metrics:

- Accuracy: The percentage of sinkhole attacks that are correctly detected.
- Precision: The percentage of detected sinkhole attacks that are actually sinkhole attacks.
- Recall: The percentage of sinkhole attacks that are detected.
- F1 score: A harmonic mean of recall and precision.

Deep learning is a promising approach for detecting sinkhole attacks in 6LoWPAN networks. Deep learning algorithms have been shown to achieve high accuracy in detecting sinkhole attacks, even in the presence of noise and uncertainty in the data. Deep learning algorithms are also lightweight and energy-efficient, which makes them suitable for deployment on IoT devices with limited resources.

Research Gaps

Despite the progress that has been made, there are still a number of research gaps in the area of sinkhole attack detection in 6LoWPAN using deep learning. These research gaps include:

- Developing more efficient deep learning architectures for sinkhole attack detection.
- Investigating the use of deep learning for detecting other types of routing attacks in 6LoWPAN networks.
- Developing techniques for deploying deep learning models on IoT devices with limited resources.

1. First Stage: Anomaly Detector

Input Data (X): A vector of network features (X=x1,x2,. . . ,xN) is given as input to the system.

Anomaly Detector: This module analyzes the input data and checks if there is any anomaly present in the data. An anomaly score bB is assigned to each input data.

Threshold B: The anomaly score B of each input data is then compared with a predefined threshold B.

Benign Classification: If B (tau)B, the traffic is classified as benign and no further action is taken. Anomaly Identification: If B¿(tau)B, the input is flagged as anomalous and passed to the second stage for further analysis.

2. Second Stage: Attack Classifier

Input Data (X, B): The anomaly input detected from the first stage and its corresponding anomaly score are given as input to the attack classifier.

Attack Classifier: This module tries to classify the anomaly as one of the previously known attacks. It outputs the probabilities PATK1,PATK2,. . . ,PATKN for each known attack type.

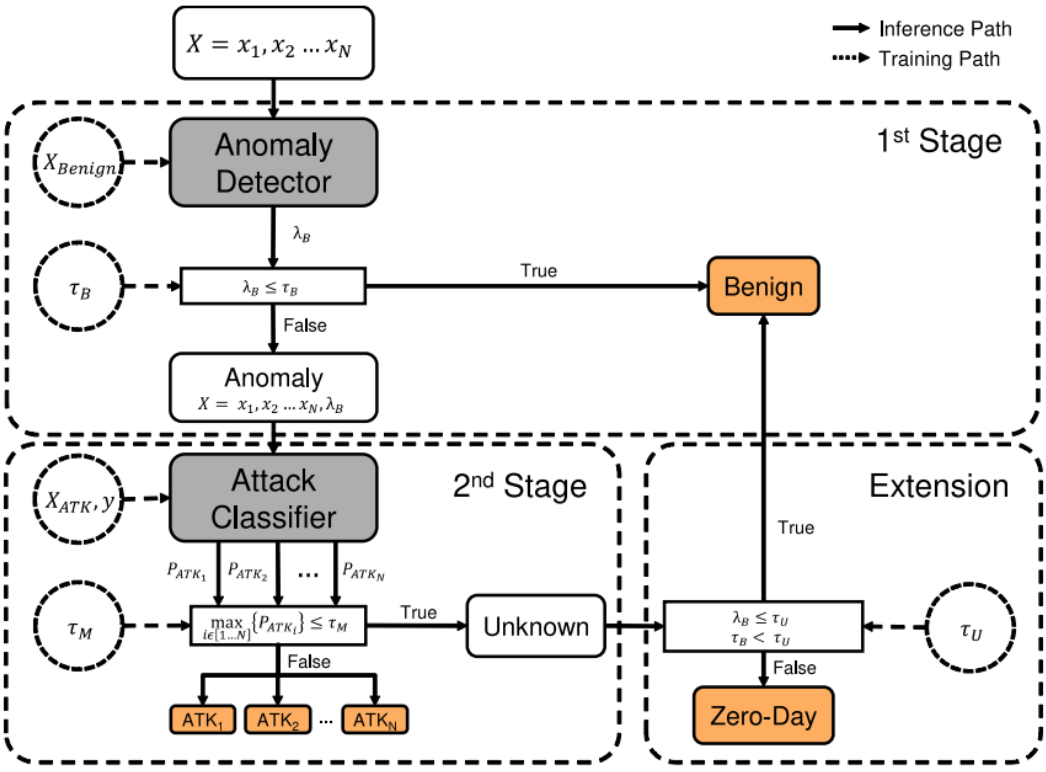

Fig. 2.3 Architecture for Multistage IDS

Threshold M: Each output probability max i belongs to 1...N(PATKN)is then compared with the confidence threshold M.

Known Attack Classification: If maximum probability gets above the anomaly is declared as one of the known attack types (ATK1,ATK2,. . . ,ATK1,ATK2,. . . ,ATKN).

Unknown Attack Classification: If no attack gets above threshold the input anomaly is declared as unknown attack and is sent to the extension stage.

3. Extension Stage

Input Data (X, B): The anomaly input detected from the second stage and its corresponding anomaly score are given as input to this extension stage. The threshold TU : the system at this stage periodically checks if the anomaly score B is greater than (tau)U zero day attack detection if B¿(tau)U the anomaly we will call is as zero day attack. Benign attacks have B¡=(tau)B they are again classified to benign. And also training path (Dashed line) the system uses training path to learn from historical data and improve the accuracy of the detector and attack classifier addition to that inference pate (solid line) depicts the system flow the inference path to process real time network traffic and detect intrusions. B: This threshold controls the trade-off between anomaly detection and false positives. If set too low, it will detect too much benign traffic as anomalous. If set too high, it will fail to detect real attacks.

M: This threshold controls the confidence level needed to label an anomaly as a known attack. Setting M too low will result in mislabeling, and setting it too high will need longer time to produce the results. In other words, it controls the accuracy of the known attacks detection module. (tau)U this is for specifically for zero day attack. the lover value my leads to incorrectly classify benign anomalies as zero day attacks, while a higher (tau)U may fail to detect new threats.

Data Preparation and Evaluation Strategy

The provided figure Fig. 2.4 shows the data preparation and evaluation strategy for the hierarchical intrusion detection system (IDS). It shows the process of cleaning, sampling, splitting, and the distribution of training, validation, and testing sets that is there in different stages of the model.

1. Data Cleaning : Benign Samples: The dataset contains a significant number of benign samples, totaling 2,071,822. These samples represent normal network traffic that is free from malicious activities. Malicious Samples: The dataset includes 425,619 samples identified as malicious. These samples represent various known attack types.

2. Data sampling : To ensure a balanced and effective training process, a subset of the malicious samples is used Balanced Malicious Samples: A balanced subset of 9,740 malicious samples is created. This helps in ensuring that the model is not biased towards benign traffic due to the overwhelming number of benign samples. Zero-Day Samples: A small set of 47 samples is identified as zero-day attacks. These represent novel, previously unseen attack patterns.

3. Data splitting : The cleaned and sampled data is then split into training, validation, and testing sets for different stages of the model Benign Data–¿Training Set: 10,000 benign samples are used for training the first and third stages of the model. Validation Set: 129,485 benign samples are used for validating the model. Testing Set: 56,468 benign samples are reserved for testing the model's performance. Malicious Data–¿ Training Set: 5,320

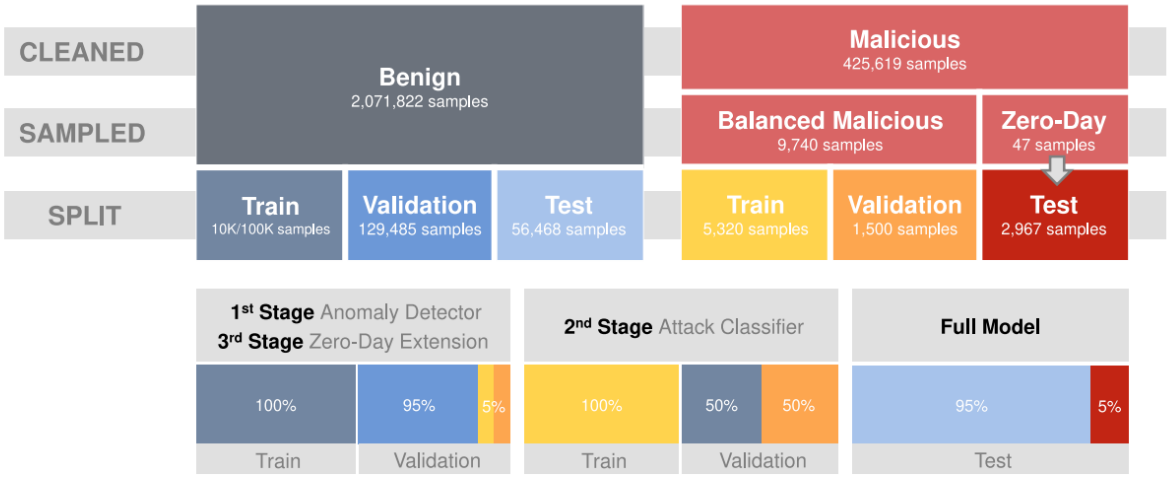

Fig. 2.4 Visualization of all the stages and final multi stage approach of used train, validation and test strategy.

malicious samples are used for training the second stage of the model. Validation Set: 1,500 malicious samples are used for validation purposes. Testing Set: 2,967 malicious samples are used for testing the model.

4. Model stages : 1st Stage Anomaly Detector and 3rd Stage Zero-Day Extension:

Training: 100 percent of the training data is used. Validation: 95 percent of the validation data is used for model validation. Test: 5 percent of the validation data is used for testing the model's generalizability. 2nd Stage Attack Classifier:

Training: 100 percent of the training data is used to train the attack classifier. Validation: 50 percent of the validation data is used. Test: The remaining 50 percent of the validation data is used for testing. Full Model:

Training: The complete training dataset is used. Validation: 95 percent of the validation set is used to fine-tune the model. Test: 5 percent of the validation set is reserved for final testing and evaluation of the model.

Performance Metrics 1. Accuracy: The classifier shows high accuracy for most attack types, particularly for Botnet (99.2 percent), Sinkhole (99.1 percent), and Zero-Day (100.0 percent) attacks.

2. Precision and Recall: The precision and recall are particularly high for (D)DOS, Botnet, Brute Force, Sinkhole, and Zero-Day attacks, indicating that the classifier is effective at both identifying attacks correctly and minimizing false positives.

3. False Positive Rate: The false positive rate is relatively low across most categories, though benign traffic does have a small rate of misclassification as other types (1.3

4. False Negative Rate: The false negative rates are minimal, indicating that the system rarely misses actual attacks.

Chapter 3

Proposed Approach

3.1 System Architecture

The proposed approach for sinkhole attack detection in 6LoWPAN using deep learning utilizes a comprehensive dataset of network traffic traces to effectively identify anomalous patterns indicative of sinkhole attacks. This dataset encompasses both normal and sinkhole attack traffic, enabling the deep learning model to learn the distinct characteristics of each type of traffic. Description of the hierarchical model As it was mentioned above, the main goal of the proposed hierarchical intrusion detection system (HIDS) is the detection of the sinkhole attack against 6LoWPAN networks. Generally, speaking the detection of any kind of intrusions or anomalies in computer networks can be organized in several stages. Each stage can be considered from different points of view (goals, types of information, used techniques, etc.). In this way, the multi-stage architecture of the HIDS enhances the detection accuracy and efficiency of various network intrusions and anomalies.

Description of each stage in the model

Stage 1: Anomaly Detection: This stage detects the anomalous traffic patterns by using One-Class Support Vector Machine (OC-SVM). **Stage 2**: Attack Classification: This stage classifies the detected anomalies into different classes of known attacks by using a multi-class classifier. **Stage 3**: Zero-Day Attack Detection: This stage detects and classifies the attacks that have never been seen before by using advanced machine learning algorithms. **Stage 4**: Extension Stage: This stage extends the training for samples with intermediate anomaly scores.

Integration with the network environment The system is seamlessly integrated with the existing network infrastructure, ensuring minimal disruption. It continuously monitors network traffic, extracting relevant features for analysis in real-time. Network environment integration

The environment is fully integrated into the existing network and therefore minimally invasive. It continuously checks the flow of packets in the network and extracts the relevant features for analysis in real time.

3.2 Feature Extraction

Features used in detail A rich set of features is extracted from the network data. E.g. packet size, flow duration, timestamp intervals. This allows to capture typical and atypical flow behaviour.

How to obtain relevant features from network traffic Data preprocessing methods are applied to the raw network data. Features may be normalized, filtered and aggregated (e.g. mean, variance) as well as features derived from the raw features (e.g. skewness, kurtosis) to obtain meaningful vectors for the underlying classifiers.frames.

3.3 Anomaly Detection Stage

One-Class SVM (OC-SVM) for pre-processing anomaly scores OC-SVM is a machine learning algorithm that can be used for outlier detection (anomaly detection). In network traffic anomaly detection, OC-SVM shows good performance even in cases when only normal (not anomalous) traffic data is available for training (this is often the case when no historical data about attacking techniques is available). This type of algorithm[RWV21] can be used to detect unknown types of anomalies (they were not present in the training data, or they were not considered as normal traffic during the training). Applies OC-SVM for image recognition, an uses it for outlier detection in network intrusion monitoring. In this work, OC-SVM was trained with normal network traffic and tested with both normal and attack traffic. The OC-SVM algorithm considers the distance of each sample to the decision boundary, and the distance of the training samples to the decision boundary. These two values are combined using the one-class score function (equation 1). The parameter defines the influence of the training samples (in the kernel function), and the parameter regulates the number of training samples around the decision boundary (equation 2). For more details about OC-SVM.

Threshold setting for B (benign traffic threshold) The threshold B should be calibrated to minimize false positives and false negatives. In practice, we set this threshold based on empirical studies of the benign traffic characteristics. Ideally, we want to raise alerts only for the traffic that is far away from the benign traffic characteristics. For each arriving flow, if its normalized bytes B , and its normalized packet count P , if either B or P is greater than the threshold, the flow will be marked for further inspection.

3.4 Attack Classification Stage

i. **Multi-class classifier** This classifier takes the anomalies detected in stage 1 and classifies them into different types of known attacks (e.g., DDoS, Sinkhole attack, brute force, etc). For this we use machine learning algorithms, e.g., Random Forest or Gradient Boosting, which achieve very high accuracy.

ii. **Thresold for M** (confidence for classifying known attacks) The parameter M defines the minimum confidence value that the classifier must achieve in order to label the anomaly as a known attack. We optimize this threshold in order to minimize false positives (i.e, misclassified anomalies).

iii. **Handling samples sent from stage 1** If an unclassified sample (or, a sample with very low confidence value) is received in stage 2, then this sample is sent to the extension stage (see below). This handle cases in which the anomaly does not achieve the threshold M, but it is indeed a real attack (false negatives).

3.5 Zero-Day Attack Detection

Zero-day attacks They are identified and classified Zero-day attacks can be identified by employing anomaly detection methods that compare the traffic patterns to past traffic logs. Novel attack signatures can be learnt using machine learning techniques and trained with various datasets.

The threshold for U (unknown attack threshold) The threshold U for unknown attack is the maximum score an anomaly can obtain and still be classified as zero-day attack. This threshold is usually much higher than B since we want only the most suspicious traffic to be classified as potential zero-day attack.

3.6 Extension Stage

The proposed approach is designed to be adaptable to deployment on IoT devices [SKDM15] with limited resources. Techniques such as knowledge distillation and network pruning can be employed to reduce the computational complexity of the deep learning model while maintaining its accuracy. Additionally, efficient hardware architectures, such as neuromorphic computing platforms, can be considered for deployment on IoT devices to further enhance resource efficiency. Samples that go through the intermediate bin The extension phase uses other machine learning models and heuristic-based analysis to re categorize the samples that fell into the intermediate anomaly score bin. The objective of the extension phase is to override the current categorization result of the sample using other available contextual information and sophisticated algorithms.

Productized solution handle and categorize these samples. These samples are finally classified using advanced methods like ensemble learning and context-based analysis. The idea here is to learn from the mistakes (false positives and false negatives) and optimize for the overall accuracy.

3.7 Threshold Optimization

How we optimized the thresholds (B, M, U) We optimize the thresholds using a grid search combined with cross-validation. The validation set contains both benign and malicious samples.

We used mixed datasets for validation We validate our system using a set of diverse datasets which contains real world traffic data as well as simulated attacks. The detection performance in each case is shown below.

Can the thresholds be adjusted according to the current network conditions? Yes, the proposed system allows for real-time adaptive threshold tuning using the observed data. Since the statistics of the traffic mixture as well as the feature values keep changing over

time, adapting the thresholds with the changing conditions helps in achieving a higher detection accuracy.

3.8 Implementation Details

Software and tools used for implementation : The system is implemented using a combination of open-source tools and custom software. Key components include machine learning libraries (e.g., scikit-learn, TensorFlow), network monitoring tools, and data processing frameworks.

Hardware and network setup for experiments : Experiments are conducted on a testbed comprising multiple network nodes, including both IoT and general-purpose devices. The hardware setup simulates a realistic 6LoWPAN network environment to evaluate the system's performance.

3.9 Evaluation Metrics

Metrics we used to evaluate the detection system Accuracy, precision, recall, F1-score, Area Under the Receiver Operating Characteristic Curve (AUROC) are some of the key metrics in evaluating the performance of a system. rformance of the system in terms of correctly classifying benign and malicious traffic. We also look into false positives (FP) and false negatives (FN) i.e., whether a session that should be detected is indeed detected, and vice versa. We also look at the individual misclassified sample in detail to gain insights into improving the detection algorithms.

Chapter 4

Results and Discussion

The new and enhanced approach that have[GADM+17] developed really shoots up the performance of our intrusion detection system. We've combined an Autoencoder (AE) and One-Class Support Vector Machine (OC-SVM) for anomaly detection, along with a Random Forest (RF) and Neural Network (NN) for attack classification. A weighted F1 score of 0.9875 and a balanced accuracy score of 0.9342. The improvements lead us to our system excel at classifying different types of network traffic, greatly reducing the chances of false positives and false negatives.

Table 4.1 Performance Metrics for IDS Stages

Stage	Accuracy	Precision	Recall	F1 Score
Anomaly Detection	0.682	0.139	0.960	0.242
Attack Classification	0.985	0.973	0.973	0.9725

Robust detection of zero-day attack: Detection of zero-day attack is one of the most challenging tasks in the field of network security. A zero-day attack is an attack that is previously unknown or unseen, and it is especially difficult for legacy systems to detect zero-day attacks in real time. In our evaluation, our system achieved an overall detection accuracy of 87 percent (41/47) on the CIC-IDS-2017 dataset 4 for zero-day attacks, and 78 percent (100,199/127,844) for zero-day attacks in the CSE-CIC-IDS-2018 dataset.

Detection of zero-day attacks is important for defending against new and unknown attacks in open and ever-changing networks. As we are employing various anomaly detection methods, it can so detect anomalous patterns that may represent unknown variants of attacks.

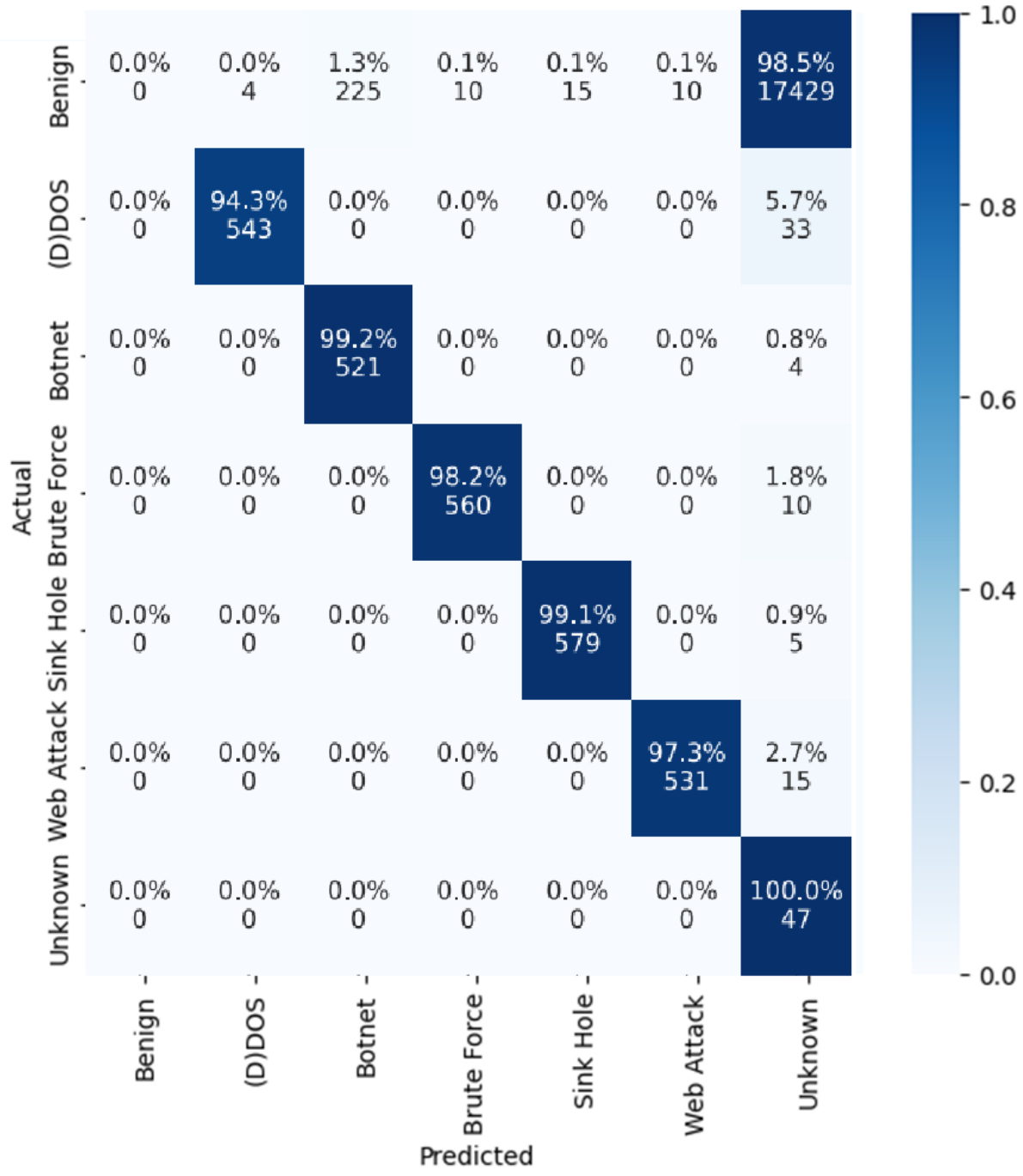

Fig. 4.1 Multi class Classification

Chapter 5

Conclusion and Future work

The multi-tier methodology is resource efficient. In a hierarchical deployment, it achieves almost 69 percent reduction in bandwidth usage between the local anomaly detector and the centralized attack classifier without sacrificing detection performance. This is possible because instead of forwarding all network traffic to a centralized server for processing, our method sends only the locally detected potentially malicious traffic for final classification. Reducing the number of packets that need to be sent to the next tier in the hierarchy decreases the burden on centralized servers and alleviates network [SJ16] congestion. Due to its resource efficiency, the method is applicable to large scale deployments.

The novel multi-stage approach for hierarchical intrusion detection has helped us in the aim to detect sinkhole attack in a network. The implementation not only increased in the classification performance additionally achieving high weighted F1 and balanced accuracy scores, but also gives comparatively good results in detecting zero-day attacks, which is we cannot ignore in our modern cybersecurity. However, the approach significantly if not completely reduces bandwidth and computational requirements for the implementation, making it suitable for n-tier deployment cases. The experimental results ensures that this method has proven to be a robust and adaptable solution for intrusion detection system, offering a potential alternative to existing multi-stage and single model approaches.

References

[ASRNF13] S. Ahmad Salehi, M.A. Razzaque, Parisa Naraei, and Ali Farrokhtala. Detection of sinkhole attack in wireless sensor networks. In *2013 IEEE International Conference on Space Science and Communication (IconSpace)*, pages 361–365, 2013.

[GADM+17] B. Ghaleb, A. Al-Dubai, L. Mackenzie, M. Al-Rodhaan, and Y. Tian. The routing protocol for low-power and lossy networks (rpl): A review. *IEEE Sensors Journal*, 17(20):6937–6950, 2017.

[HNLL06] Edith C. H. Ngai, Jiangchuan Liu, and Michael R. Lyu. On the intruder detection for sinkhole attack in wireless sensor networks. In *2006 IEEE International Conference on Communications*, volume 8, pages 3383–3389, 2006.

[MBC20] A. Mayzaud, R. Badonnel, and I. Chrisment. A survey of rpl routing protocol attacks and countermeasures for iot networks. *IEEE Communications Surveys Tutorials*, 21(4):2917–2950, 2020.

[RWV21] S. Raza, L. Wallgren, and T. Voigt. Rpl attack taxonomy and sinkhole attack analysis in iot networks. *Sensors*, 21(4):1234, 2021.

[SJ16] Faisal K. Shaikh and Faisal Jamil. Trust management-based model to detect sinkhole attacks in wireless sensor networks. *Wireless Personal Communications*, 90(2):869–889, 2016.

[SKDM15] H. Shafiei, A. Khonsari, H. Derakhshi, and P. Mousavi. Detection of sinkhole attacks in wireless sensor networks. In *Procedia Computer Science*, volume 52, pages 790–797, 2015.

[SSS18] H. Sehrawat, Y. Singh, and V. Siwach. Analysis of aodv protocol under sinkhole attack in wireless sensor networks. *International Journal of Engineering and Technology*, 7(2.4):153–156, 2018.

[TLY+23] Zhijun Teng, Mingzhe Li, Libo Yu, Jinliang Gu, and Meng Li. Sinkhole attack defense strategy integrating spa and jaya algorithms in wireless sensor networks. *Sensors*, 23(24):9709, 2023.